```
NNN       NNN  MMM        MMM  LLL
NNN       NNN  MMM        MMM  LLL
NNN       NNN  MMM        MMM  LLL
NNN       NNN  MMMMMM  MMMMMM  LLL
NNN       NNN  MMMMMM  MMMMMM  LLL
NNN       NNN  MMMMMM  MMMMMM  LLL
NNNNNN    NNN  MMM  MMM    MMM  LLL
NNNNNN    NNN  MMM  MMM    MMM  LLL
NNNNNN    NNN  MMM  MMM    MMM  LLL
NNN  NNN  NNN  MMM        MMM  LLL
NNN   NNN NNN  MMM        MMM  LLL
NNN   NNN NNN  MMM        MMM  LLL
NNN    NNNNNNN MMM        MMM  LLL
NNN    NNNNNN  MMM        MMM  LLL
NNN    NNNNNN  MMM        MMM  LLL
NNN       NNN  MMM        MMM  LLL
NNN       NNN  MMM        MMM  LLL
NNN       NNN  MMM        MMM  LLL
NNN       NNN  MMM        MMM  LLLLLLLLLLLLLLL
NNN       NNN  MMM        MMM  LLLLLLLLLLLLLLL
NNN       NNN  MMM        MMM  LLLLLLLLLLLLLLL
```

_$

Ps
--
NP

NP

$G

$O

NP

PA

_L

```
NN      NN PPPPPPP      AAAAAA   MM      MM   AAAAAA      CCCCCCCC
NN      NN PPPPPPP      AAAAAA   MM      MM   AAAAAA      CCCCCCCC
NN      NN PP      PP AA      AA MMMM  MMMM AA      AA CC
NN      NN PP      PP AA      AA MMMM  MMMM AA      AA CC
NNNN    NN PP      PP AA      AA MM MM MM MM AA      AA CC
NNNN    NN PP      PP AA      AA MM MM MM MM AA      AA CC
NN NN   NN PPPPPPP    AA      AA MM  MM  MM AA      AA CC
NN  NN  NN PPPPPPP    AA      AA MM      MM AA      AA CC
NN   NNNN  PP         AAAAAAAAAA MM      MM AAAAAAAAAA CC
NN   NNNN  PP         AAAAAAAAAA MM      MM AAAAAAAAAA CC
NN     NN  PP         AA      AA MM      MM AA      AA CC       ....
NN     NN  PP         AA      AA MM      MM AA      AA CC       ....
NN     NN  PP         AA      AA MM      MM AA      AA CCCCCCCC ....
NN     NN  PP         AA      AA MM      MM AA      AA CCCCCCCC ....


MM      MM   AAAAAA   RRRRRRR
MM      MM   AAAAAA   RRRRRRR
MMMM  MMMM AA      AA RR      RR
MMMM  MMMM AA      AA RR      RR
MM MM MM MM AA      AA RR      RR
MM MM MM MM AA      AA RR      RR
MM      MM AA      AA RRRRRRR
MM      MM AA      AA RRRRRRR
MM      MM AAAAAAAAAA RR  RR
MM      MM AAAAAAAAAA RR  RR
MM      MM AA      AA RR    RR
MM      MM AA      AA RR    RR
MM      MM AA      AA RR      RR
MM      MM AA      AA RR      RR
```

```
; Version:       'V04-000'
;******************************************************************
;*                                                                *
;*   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                      *
;*   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.       *
;*   ALL RIGHTS RESERVED.                                         *
;*                                                                *
;*   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;*   ONLY IN ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
;*   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
;*   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
;*   OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
;*   TRANSFERRED.                                                 *
;*                                                                *
;*   THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
;*   AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
;*   CORPORATION.                                                 *
;*                                                                *
;*   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;*   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.      *
;*                                                                *
;******************************************************************

.MACRO  NPADF$
;
; NPARSE function code definitions
;
        NPA$_IMAGE = 0
        NPA$_WORD = 1
        NPA$_MASK = 2
        NPA$_BYTE = 3
        NPA$_EOM = 4
        NPA$_NULL = 5
        NPA$_SBEXP = 6
        NPA$_ERROR = 7
        NPA$_MATCH = 8
        NPA$_LOOK = 9
        NPA$_EXTZV = 10
;
; NPARSE matching codes
;
        NPA$_EXIT = 0           ; Done
        NPA$_FAIL = -1          ; Fail the parse
        NPA$_ADVANCE = 1        ; Advance pointers
;
; NPARSE flag bit definitions
;
        NPA$M_EXT        = ^X01
        NPA$M_PARAM      = ^X02
        NPA$M_ACTION     = ^X04
        NPA$M_STATE      = ^X08
        NPA$M_MASK       = ^X10
```

```
        NPA$M_MSKADR    = ^X20
        NPA$M_OFFSET    = ^X40
        NPA$M_LAST      = ^X8000
.ENDM   NPADF$
.MACRO  IMSG$,STTBL,DEBUG
        .PSECT  NPA$STATE,NOWRT,NOEXE,BYTE
STTBL::
        .IF     IDENTICAL       <DEBUG>,<$DEBUG>
        .MACRO  DBGTP$
        .LIST
                                .=.
        .NLIST
        .ENDM
        .IF_FALSE
        .MACRO  DBGTP$
        .ENDM
        .ENDC

        NPADF$
        FLG$$$ = -1
        NXT$$$ = 0
        .PSECT
.ENDM   IMSG$
.MACRO  FIELD$,LABEL
        .PSECT  NPA$STATE
        FLG$$$ = FLG$$$!NPA$M_LAST
        NTRAN$
        .IF     NOT_BLANK       <LABEL>
LABEL:
        .ENDC
        FLG$$$ = -1
        .PSECT
.ENDM   FIELD$
.MACRO  NTRAN$
        .PSECT  NPA$STATE
        .IF     EQUAL   FLG$$$+1
        FLG$$$ = 0
        .MEXIT
        .ENDC
        TYP$$$
        .WORD   FLG$$$
        .IF     NOT_EQUAL       FLG$$$&NPA$M_EXT
        EXT$$$
        .ENDC
        .IF     NOT_EQUAL       FLG$$$&NPA$M_PARAM
        PRM$$$
        .ENDC
        .IF     NOT_EQUAL       FLG$$$&NPA$M_ACTION
        ACT$$$
        .ENDC
        .IF     NOT_EQUAL       FLG$$$&NPA$M_MASK
        BIT$$$
        .ENDC
        .IF     NOT_EQUAL       FLG$$$&NPA$M_STATE
        STA$$$
        .IF_FALSE
```

```
          .IF     EQUAL   FLG$$$&NPA$M_LAST
          .IF     EQUAL   NXT$$$
          .ERROR  ; BAD DEFAULT TRANSITION
          .ENDC
          .ENDC
          .ENDC
          FLG$$$ = 0
          NXT$$$ = 0
.ENDM     NTRAN$
.MACRO    ITRAN$,LABEL,ACTION,MASK,ADDR,PARAM
          .IF     NOT BLANK       <PARAM>
          FLG$$$ = FLG$$$!NPA$M_PARAM
          .MACRO  PRM$$$
          .ADDRESS        PARAM
          .ENDM   PRM$$$
          .ENDC
          .IF     NOT BLANK       <ACTION>
          FLG$$$ = FLG$$$!NPA$M_ACTION
          .MACRO  ACT$$$
          .ADDRESS        ACTION
          .ENDM   ACT$$$
          .ENDC
          .IF     NOT BLANK       <MASK>
          FLG$$$ = FLG$$$!NPA$M_MASK!NPA$M_MSKADR
          .IF     BLANK   <ADDR>
          .ERROR  ; MASK ADDRESS NOT PRESENT
          .ENDC
          .MACRO  BITS$$$
          .LONG   MASK
          .ADDRESS        ADDR
          .ENDM   BITS$$$
          .ENDC
          .IF     NOT BLANK       <LABEL>
          FLG$$$ = FLG$$$!NPA$M_STATE
          .MACRO  STA$$$
          .ADDRESS        LABEL
          .ENDM   STA$$$
          .ENDC
.ENDM     ITRAN$
.MACRO    $IMAGE,SIZE,LABEL,ACTION,MASK,ADDR,PARAM
          .PSECT  NPA$STATE
          NTRAN$
          DBGTP$
          .MACRO  TYP$$$
          .WORD   NPA$ IMAGE
          .ENDM   TYP$$$
          .MACRO  EXT$$$
          .LONG   SIZE
          .ENDM   EXT$$$
          FLG$$$ = FLG$$$!NPA$M_EXT
          ITRAN$  LABEL,ACTION,MASK,ADDR,PARAM
          .PSECT
.ENDM     $IMAGE
.MACRO    $WORD,VALUE,LABEL,ACTION,MASK,ADDR,PARAM
          .PSECT  NPA$STATE
          NTRAN$
```

```
        DBGTP$
        .MACRO  TYP$$$
        .WORD   NPA$_WORD
        .ENDM   TYP$$$
        .MACRO  EXT$$$
        .ADDRESS        VALUE
        .ENDM   EXT$$$
        FLG$$$ = FLG$$$!NPA$M_EXT
        ITRAN$  LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT
.ENDM   $WORD
.MACRO  $BYTE,VALUE,LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT  NPA$STATE
        NTRAN$
        DBGTP$
        .MACRO  TYP$$$
        .WORD   NPA$_BYTE
        .ENDM   TYP$$$
        .MACRO  EXT$$$
        .ADDRESS        VALUE
        .ENDM   EXT$$$
        FLG$$$ = FLG$$$!NPA$M_EXT
        ITRAN$  LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT
.ENDM   $BYTE
.MACRO  $EOM,DUMMY,LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT  NPA$STATE
        NTRAN$
        DBGTP$
        .MACRO  TYP$$$
        .WORD   NPA$_EOM
        .ENDM   TYP$$$
        ITRAN$  LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT
.ENDM   $EOM
.MACRO  $MASK,BITMSK,LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT  NPA$STATE
        NTRAN$
        DBGTP$
        .MACRO  TYP$$$
        .WORD   NPA$_MASK
        .ENDM   TYP$$$
        .MACRO  EXT$$$
        .LONG   BITMSK
        .ENDM   EXT$$$
        FLG$$$ = FLG$$$!NPA$M_EXT
        ITRAN$  LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT
.ENDM   $MASK
.MACRO  $SBEXP,RTN,LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT  NPA$STATE
        NTRAN$
        DBGTP$
        .MACRO  TYP$$$
        .WORD   NPA$_SBEXP
        .ENDM   TYP$$$
```

```
        .MACRO  EXT$$$
        .ADDRESS          RTN
        .ENDM   EXT$$$
        FLG$$$ = FLG$$$!NPA$M_EXT
        ITRAN$  LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT
.ENDM   $SBEXP
.MACRO  $NULL,DUMMY,LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT  NPA$STATE
        NTRAN$
        DBGTP$
        .MACRO  TYP$$$
        .WORD   NPA$_NULL
        .ENDM   TYP$$$
        ITRAN$  LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT
.ENDM   $NULL
.MACRO  $LOOK,VALUE,LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT  NPA$STATE
        NTRAN$
        DBGTP$
        .MACRO  TYP$$$
        .WORD   NPA$_LOOK
        .ENDM   TYP$$$
        .IF     NOT_BLANK       <VALUE>
        .MACRO  EXT$$$
        .LONG   VALUE
        .ENDM   EXT$$$
        FLG$$$ = FLG$$$!NPA$M_EXT
        .ENDC
        ITRAN$  LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT
.ENDM   $LOOK
.MACRO  $EXTZV,FLD,LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT  NPA$STATE
        NTRAN$
        DBGTP$
        .MACRO  TYP$$$
        .WORD   NPA$_EXTZV
        .ENDM   TYP$$$
        LIST$   FLD
        FLG$$$ = FLG$$$!NPA$M_EXT
        ITRAN$  LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT
.ENDM   $EXTZV
.MACRO  LIST$,VAL,POS,SIZ,FLG
        .MACRO  EXT$$$
        .IF     NOT_BLANK       VAL
        .BYTE   VAL     ; Field value
        .IF FALSE
        .ERROR  ; $EXTZV - FIELD VALUE NOT SPECIFIED
        .ENDC
        .IF     NOT_BLANK       POS
        .BYTE   POS     ; Field position
        .IF FALSE
        .ERROR  ; $EXTZV - FIELD POSITION NOT SPECIFIED
```

```
        .ENDC
        .IF     NOT_BLANK       SIZ
        .BYTE   SIZ     ; Field size
        .IF_FALSE
        .ERROR  ; $EXTZV - FIELD SIZE NOT SPECIFIED
        .ENDC
        .IF     NOT_BLANK       FLG
        .BYTE   FLG     ; Flags
        .IF_FALSE
        .BYTE   0       ; No flags
        .ENDC
        .ENDM   EXT$$$
.ENDM   LIST$
.MACRO  $ERROR,ERR,LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT  NPA$STATE
        NTRAN$
        DBGTP$
        .MACRO  TYP$$$
        .WORD   NPA$_ERROR
        .ENDM   TYP$$$
        .MACRO  EXT$$$
        .LONG   ERR
        .ENDM   EXT$$$
        FLG$$$ = FLG$$$!NPA$M_EXT
        ITRAN$  NPA$_FAIL,ACTION,MASK,ADDR,PARAM
        .PSECT
.ENDM   $ERROR
.MACRO  $MATCH,NBR,LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT  NPA$STATE
        NTRAN$
        DBGTP$
        .MACRO  TYP$$$
        .WORD   NPA$_MATCH
        .ENDM   TYP$$$
        .MACRO  EXT$$$
        .LONG   NBR
        .ENDM   EXT$$$
        FLG$$$ = FLG$$$!NPA$M_EXT
        ITRAN$  LABEL,ACTION,MASK,ADDR,PARAM
        .PSECT
.ENDM   $MATCH
.MACRO  $NEXT
        NXT$$$ = -1
        NTRAN$
        DBGTP$
        FLG$$$ = -1
.ENDM   $NEXT
```

CNFSEND
LIS

NPADEF
MDL

CNFSTORE
LIS

NMLSHR
MAP

NMAHEAD
B32

NPAMAC
MAR

CNFSHOW
LIS

CNFWQDEF
LIS

NMATAIL
B32

NML

NMLDDL
B32

NML
MAP

NMLDEF
MDL

NMAFILES
LIS

NMAFIELDS
LIS

CNFWORKQ
LIS